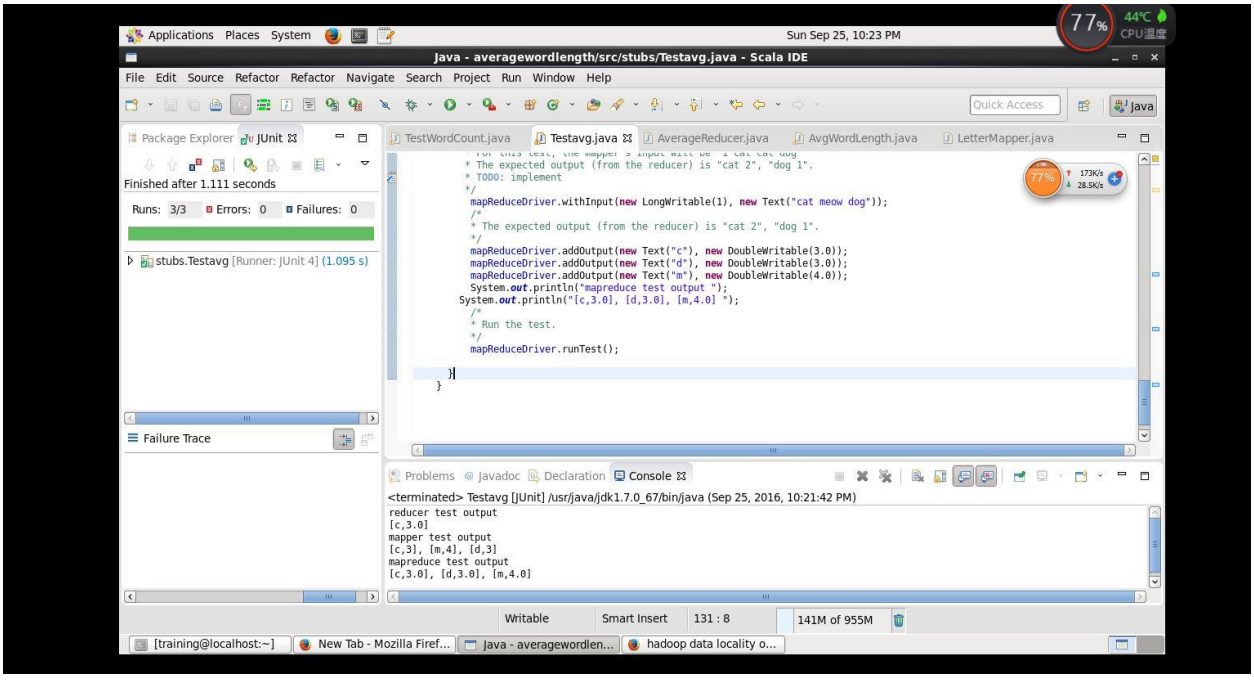
Michael Wang michaelwang 438275 Terry Lyu terrylu 435091
HW 3

1a. Yes, because since we are not using combiners, some reduce functions will be on keys (words) that occur extremely frequently and thus have a long list of values, while some reduce functions will be on words that only occur occasionally and will finish quickly.

1b. If there were only 10 reduce tasks we would not expect there to be great skew because we are randomly distributing the keys among the reduce tasks and would expect that with so many keys for each task to process, they would all average out to take a similar amount of time. With 10,000 Reduce tasks we would expect there to be skew because each task would only get a small number of keys and since some words occur much more often than others, it is impossible to distribute the work evenly. Imagine if each Reduce task only got one key. Then, one of the 10,000 Reduce tasks may get a very obscure word like “xylophone” that only occurs once, while another would get the word “the”, which would occur a large number of times.

2. Data locality in reference to Hadoop refers to how map reduce tasks are executed on the node that the data is stored on. Essentially your input file/s are distributed using HDFS and placed on multiple nodes, then the master node assigns the map function to be called on the node that the file is stored on. This is more efficient because you don’t have to transfer your input file to the node that is going to execute the map function. Then, the intermediate file output from the map task is saved on the same node, again saving time and resources. If data locality is not possible, then the file has to be transferred to the node that is working on it, slowing down the entire process.

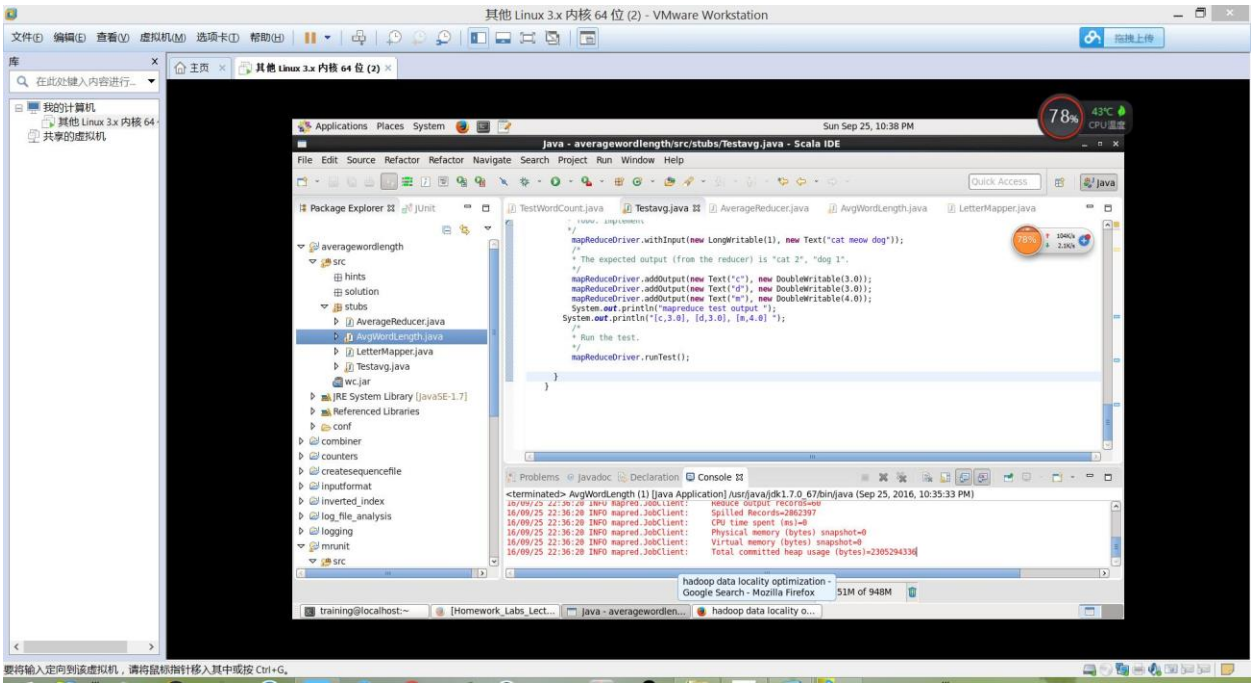
3b.



4a. When testing MapReduce programs locally the main differences are that your input and output location are on your physical hard drive, rather than HDFS. Additionally print statements for both stderr and stdout will appear in your console when running locally, but only stderr statements will be shown for programs run on Hadoop. Also, it is easier to make incremental changes when you testing locally because it’s quicker and more responsive. Last, you can kill jobs using ctrl+c locally, but on Hadoop you have to kill by job id.

Michael Wang michaelwang 438275 Terry Lyu terrylu 435091
HW 3

4b.




```
Configuration conf = new Configuration();
conf.set("fs.defaultFS", "file:///");
conf.set("mapreduce.framework.name", "local");

/*
 * Instantiate a Job object for your job's configuration.
 */
Job job = new Job(conf);
```

4c.

4d. I would prefer to modify the java file and then run directly on command line.

Reason: 1. The file system on our local computer is way easier to navigate than HDFS

2. after modifying the program it's easier to change the input and output location because we can simply change it before executing the job.

3. if we choose not to run it locally afterwards, we can simply comment out our code in the program

4e. After testing a program locally, you want to test it in pseudo-distributed mode with larger amounts of representative sample data to cover all test cases. Next, run it on the cluster with small amounts of data to make sure that it will run on the cluster without breaking before you commit to running it on all of your data. Finally, run the application with all of the data on the cluster.